RESSP:基于FPGA的可重构SDN交换结构

何璐蓓 厉俊男 杨翔瑞 孙志刚
(国防科技大学计算机学院 长沙 410073)

摘 要 SDN采用转发与控制分离的架构和集中的控制管理机制,可有效满足不同网络中不同粒度的管理控制需求。当高校科研人员进行SDN的教学和创新实验时,需要一个处理过程可感且可重新编程的数据平面来支持原理展示和自主研究。然而,传统ASIC交换机的内部实现流程不透明且转发查表架构固定,软件交换机的处理性能较低,因此无法充分支持数据平面的研究。目前,通过FPGA设计可编程数据平面,为满足不同科研场景下多样化的处理需求提供了一条可行路径。但是,在基于FPGA的可重构交换机架构和设计方法方面还缺少深入研究,主要表现在难以实现基于模块细粒度的SDN处理流程重构,现有工作复用程度低,同时无法为开源的SDN数据平面设计提供技术支持。为此,提出一种基于FPGA的SDN交换平面实现结构——RESSP(FPGA-based REconfigurable SDN Switching Pipeline)。RESSP将报文处理流程拆解成多个可动态加载的模块,针对交换机具体的应用场景,利用FPGA可编程特性对硬件功能模块进行增加、删除或替换,从而针对实际需求设计出相应的报文处理逻辑。此外,基于RESSP实现了一个SDN交换机的原型系统MiniSwitch。MiniSwitch验证了RESSP在教学科研实验中快速重构所需SDN数据平面的可行性和可扩展性。

关键词 软件定义网络,现场可编程门阵列,交换结构,可重构,开源,网络教学
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.01.036

RESSP:An FPGA-based REconfigurable SDN Switching Architecture

HE Lu-bei LI Jun-nan YANG Xiang-rui SUN Zhi-gang
(College of Computer Science,National University of Defense Technology,Changsha 410073,China)

Abstract SDN,which uses forwarding control separation architecture and centralized management control mechanism,can effectively meet the needs of different networks in different granularity control demand. When SDN teaching and innovation experiments are carried out by researchers in universities,a data plane is needed which can be felt and reprogrammed to support the principle demonstration and the independent research. However,the internal implementation process of traditional ASIC switch is opaque and the lookup architecture is fixed,and the processing speed of the software switch is low,so they can not fully support the research of the data plane. At present,the design of programmable data plane whith FPGA provides a feasible path to meet the diverse needs of different research scenarios. Although academia and industry have been done some preliminary attempt based on FPGA SDN switch design,but FPGA-based reconstructed switch architecture and design method still lack in-depth study,and it is difficult to achieve fine-grained module reconfigurable SDN processing. Therefore,the existing work is hard to reuse and is also unable to provide technical support to SDN data graphic design. This paper proposed a FPGA-based reconfigurable SDN switching architecture,namely RESSP. RESSP disassembles the packet processing into multiple modules which can be dynamically loaded. For specific application scenarios switches,a corresponding packet processing was designed by using FPGA to add,remove or replace the RESSP's module. Based on the structure of RESSP,this paper implemented a prototype of SDN switch MiniSwitch and its management software. MiniSwitch verifies that RESSP can quickly reconstruct the corresponding SDN data plane for different scenarios,and meet the diverse processing requirements of SDN switches in different application scenarios.

Keywords Software defined networking,FPGA,Switching architecture,REconfigurable,Open source,Network teaching

1 绪论

软件定义网络[1](Software Defined Networking,SDN)基于标准转发层抽象,采用转发与控制分离的架构和集中的控制管理机制,可有效满足不同网络中不同粒度的管理控制需求,是目前被广泛认可、影响力最大的一种新型网络体系架构。随

到稿日期:2016-12-05 返修日期:2017-03-31 本文受国家高技术研究发展计划:IPv6大规模编址与路由关键技术研究和验证(SS2015AA010201)资助。

何璐蓓(1994—),女,硕士生,主要研究方向为计算机网络,E-mail:conniemehlb@163.com(通信作者);厉俊男(1992—),男,博士生,主要研究方向为网络处理器、体系结构,E-mail:nudt_ljn@163.com;杨翔瑞(1993—),男,硕士生,主要研究方向为计算机网络,E-mail:18635723769@163.com;孙志刚(1973—),男,博士,研究员,CCF会员,主要研究方向为计算机网络体系结构、高性能路由器,E-mail:sunzhigang@263.net。

着 SDN 技术理论的不断发展与成熟，SDN 不仅代表着网络工业的发展方向，也是目前学术界最热门的研究课题之一。

作为数据转发面，交换机是 SDN 网络的核心设备，是实施网络控制策略的载体，承担着报文的转发处理工作，其设计直接对性能表现起决定性作用。随着组网方式的日趋不同及协议类型的不断增多，交换机需要支持更多的功能，设计变得越来越复杂。

与开源控制器软件呈现出的百花齐放的发展态势不同，现有的数据平面无法充分支持高校科研人员对 SDN 技术的探索和实践。传统基于 ASIC 芯片实现的交换机的内部实现流程不透明，且转发查表架构固定，不能根据应用场景和具体科研需求进行更改，无法满足可编程网络的需求。虽然已有研究提出了可编程交换 ASIC 的设计方案[2-3]，但其设计成本很高，无法实现不依赖于 Match-Action 模式的有状态处理功能，不适用于教学科研。软件交换机的成本低，功能完善，配置也比较灵活，但其性能无法满足稍大规模的实验网络的要求。

通过 FPGA(Field Programmable Gate Array)设计可编程数据平面，为满足不同科研场景下多样化的处理需求提供了一条可行路径。FPGA 可无限重复编程，利用重新配置减少了硬件的开销，灵活性极佳；FPGA 不需要预先全面设计整个架构，可以及时修改 RTL(Register Transfer Level)，同时继续进行设计活动，并针对性能和规模进行优化，极大地缩短了开发周期。虽然 FPGA 的性能与 ASIC 还存在一定差距，但随着工艺的进步，FPGA 的性能和规模在不断提升，工业界已经推出了基于 FPGA 的 SDN 交换机产品原型[4]，学术界也基于 FPGA 板卡设计了多款 SDN 交换机原型[5-7]。

然而，目前关于可重构 SDN 数据平面设计的研究还比较欠缺。虽然学术界都将相关研究开源[5-7]，但仅限于公开设计代码，并没有对如何进行模块级的重构展开深入研究，很难支持已有设计的重复使用，同时第三方按需进行 SDN 交换平面的定制也是基于这些项目的开源社区难以发展的原因。

本文提出一种基于 FPGA 的开源可重构交换结构 RESSP。RESSP 将报文处理流程拆解成多个独立的报文处理阶段，每个阶段由一个或者多个模块实现，并分别为每个报文处理阶段建立相应的模块库，以实现灵活的修改、替换等操作。RESSP 允许使用者根据实际使用需求自由选择需要的处理模块，以用于快速重构报文处理流水线。开发者在对 SDN 交换机设计并增加新功能时，也不再需要重新构建报文处理流程和代码结构，而是按照 RESSP 的设计规范在模块库中增加、选择新的模块，并使用所提供的标准接口与其他模块相连，从而大幅降低了网络应用服务开发的难度并缩短了网络设备的开发周期。

基于 RESSP 架构，在 NetMagic 硬件[8]上实现了 SDN 交换机原型系统 MiniSwitch，并在 GitHub 上建立了以 RESSP 为硬件开发方法的 SDN 交换机开源项目 FAST(FPGA-bAsed SDN swiTch)[9]，以支持 SDN 技术爱好者进行交流研究，通过移植改良或增加应用进一步推动了 SDN 交换技术的发展。

本文的主要工作和创新点包括：

(1)提出了模块化的 RESSP 结构及其流水实现方法，介绍了工作原理和流程，为实现可重构的 SDN 数据平面设计提供了模板。

(2)基于 NetMagic 硬件实现了典型的 RESSP 结构的应用——SDN 交换机原型系统 MiniSwitch。初步验证了 RESSP 结构的有效性，为科研人员针对不同场景设计不同的 SDN 处理流程提供了样例。

本文第 2 节介绍了 SDN 交换平面设计及开源的相关研究；第 3 节详细描述了 RESSP 流水线的设计思想和工作模型；第 4 节介绍了基于 RESSP 的 SDN 交换机原型 MiniSwitch；第 5 节对 RESSP 进行了进一步讨论；最后对全文进行总结和展望。

2 相关研究

为推动 SDN 交换技术的发展，使交换机能够灵活适应不同的应用场景，达到网络可编程的目标，研究人员做出了多方面的尝试和努力。

2.1 转发平面能力描述方法

ONF 组织下的转发抽象工作组(FAWG)于 2012 年 8 月提出了名为 TTP(Table Typing Patterns)[10]的数据平面支持 SDN 能力的描述方法，用以刻画现有 ASIC 芯片基于已有固定的逻辑和表组合出支持 OpenFlow 转发的能力。TTP 的实质是为控制器及网络配置人员提供一个抽象的 OpenFlow 逻辑交换机视图，允许开发者基于实际的应用需求和现有的芯片架构来定义多种转发模型。Google 在用 SDN 改造全球性广域网路架构的 B4 案例中采用的就是 TTP，在传统的交换芯片上做相应的优化包装，在接口上遵循 OpenFlow 的要求，即能满足大多数应用的需求[11]。

斯坦福大学研究人员定义了 P4[12]高级语言，其有望演变成支持 OpenFlow2.0 的高级语言编程标准。P4 所采用的编程模式允许转发流程的动态重配置，先通过配置阶段采用有向图的方式定义协议解析过程，包括转发流程所涉及的各个流表、流表之间的逻辑关系以及各个流表匹配后可能执行的动作集合；接着进行运行时的流表控制，主要任务是对流表表项的下发、修改、删除及表项匹配动作的选择[13]。Barefoot 公司于 2016 年推出了可编程交换机 Tofino[14]，其使用 P4 语言进行编程，可以支持灵活的网络协议和转发规则，力图让运营商可以用像改变软件一样快的速度来改变网络行为。

TTP 定义了一种“自底向上”的交换设计方法，即芯片厂商首先通过 TTP 规范定义自己可以支持哪些流表、流表的规模和处理动作，SDN 软件开发人员再根据这些描述开发相应的控制软件，而这种经济折衷式的方案无法带来新的功能。而 P4 追求的是“从上到下”彻底的可编程化，即网络设计者和软件开发人员根据应用场景提出对 SDN 数据平面的需求，然后数据平面实现者再将这些需求落实到具体的实现机制上。但将高级语言编译到 ASIC 芯片的指令集上依旧受到很多限制，无法完成更多复杂计算的网络功能，比如网络数据传输中自定义的加解密算法等。

2.2 可编程流水线

交换流水线是 SDN 交换机数据平面设计的核心，通常包含报文分析、查表和转发动作执行等环节。斯坦福大学提出的可扩展的 SDN 交换必须支持可编程的数据包解析提取模块[15]，即可配置的报文解析器是实现交换功能扩展，特别是支持各种不断涌现的新网络协议的前提。

Pat Bosshart 等人提出了 RMT(Reconfigurable Match

Tables)模型[2]，其允许在无须修改硬件的情况下，转发平面能修改、增加匹配表的字段，支持任意深度和宽度的流表。RMT 能够使硬件解决数据层的转发规则匹配过于严格和动作集元素数量太少等限制性问题，但这一方案十分依赖存储资源，不适合 FPGA 实现。

Intel 于 2012 年发布了 FM6000 系列混合以太网交换机[3]，采用名为 FlexPipe 的流水线处理架构来支持 OpenFlow 协议，同时也能处理传统交换和路由协议。解析模块采用 TCAM/SRAM/MUX 结构，匹配模块采用 TCAM/SRAM/LOGIC 结构，能够满足 IPv6、NAT、负载均衡、QoS、流量整形等数据中心常用的需求。但是 TCAM 的成本和功耗都很高，无法大规模部署，这必然会成为芯片设计的瓶颈。

基于 FPGA 的 SDN 转发平面设计思想与上述可编程流水线不同。FPGA 可重构的方法是选择应用场景需要的逻辑，去除不必要的逻辑，因此设计简洁，针对性强。

2.3　基于 FPGA 的 SDN 交换机实现

在工业界，SDN 初创公司 Corsa 抛弃了 ASIC 和网络处理器，选择 FPGA 来开发交换机的系统架构[4]，其基本理念是基于 FPGA 的重构特性，设计规模适合部署、具有超高性能且可以对其进行调整和修改以满足特定网络需求的 SDN 交换机。该公司于 2014 年发布了两种型号的 SDN 交换机[16]。Xilinx 公司于 2014 年推出了 SDNet 软件定义规范环境（Software Defined Specification Environment for Network）[17]，SDNet 结合使用 FPGA 和 SoC(System on Chip)，将可编程能力和智能化功能从控制层扩展至数据层。评论机构 Linley Group 表示："SDNet 软件定义规范环境能用高层次描述创建和调整网络元素，其适用范围绝不限于基于 OpenFlow 协议的 SDN[18]。"目前，Xilinx 在全球范围内开始试点并使用 SDNet 的技术研究。

在学术界，斯坦福大学基于 NetFPGA 实现了硬件加速的线速 OpenFlow 交换机[5]，其兼顾了性能和可编程性，可用于教学和网络实验。但是它由于实现较早，仅能支持 OpenFlow1.0 协议，后续发展停滞；而且 NetFPGA 是 PCI 接口卡形态，部署时必须安装在宿主机上，从而限制了应用范围。

国内在基于 FPGA 的 SDN 交换机设计方面也有一定的研究工作。南京叠锶公司基于 Zynq 器件开发了一个完全可编程的 SDN 交换机 ONetSwitch[6]。ONetSwitch 结合了 ARM 处理器的软件可编程及 FPGA 的硬件可编程，实现了软、硬件的加速 SDN 原型的设计和部署。利用多个 ONetSwitch互联实现了名为 DesktopDC 的数据中心网络试验台，尺寸小且功耗低。

国防科技大学于 2012 年基于 NetMagic08 实现了 OpenFlow1.0 交换机[7]，并用视频转发应用来验证模型的有效性；2015 年，在 NetMagic Pro 平台上基于多核与 FPGA 相结合的方式实现了支持 OpenFlow1.3 的交换机 OFS-Pro，但该平台上的 OpenFlow 多级转发流水线基于多核上的 OVS(OpenVSwitch)软件实现，FPGA 硬件仅实现了部分端口功能的加速，性能难以进一步提升。

3　RESSP 架构

3.1　研究动机

在 SDN 的教学工作中，学生们需要关心交换机的内部原理，直观地了解报文的处理流程以及南向接口与控制器的通信流程，而不仅仅是通过外部操作命令来学会如何运行网络设备。在 SDN 科研实验里，研究人员需要在交换机中增加自己的创新工作或者修改代码来实现定制实验床。而商用的 ASIC 交换机就像一个"黑盒子"，研究人员无法获得处理过程的细节并清晰展示详细的工作流程，同时也不能进行模块的增改，无法在硬件处理层次上进行创新实验。

对于学术界的科研人员来说，需要一个内部处理流程可见、处理功能可定制的数据平面。因此，基于 FPGA 设计可编程数据平面成为了可同时满足网络硬件教学和创新实验设计的最佳选择。

对于报文处理流程而言，需要按照功能划分差异性，才能根据实际需求有的放矢，进行工作量最小的修改与优化。

与此同时，技术的快速发展促使交换机快速更新，仅网络虚拟化技术就引入了许多新的报文格式和处理方式，VxLAN 和 NVGRE 等不断涌现的新型协议也给交换机的功能升级带来了巨大挑战。

对于同一个数据类型的报文，SDN 分组交换依据的字段是不同的，因此匹配表项会产生较大的差异。同样，处理一个 IPv4 报文时，防火墙应用关注的是五元组（协议类型，源、目的 IP 地址，源、目的端口），保障服务质量的应用还需要提取区分服务字段，网络测量会特别关注时间戳字段。

流量监测、网络安全和 QoS 等增值网络服务也对交换机处理报文方式的多样性提出了更高的要求。传统的仅支持简单转发或丢弃的交换机无法充分满足新的网络服务需求。以 QoS 为例，该服务可能会要求简单地增加报文头部的 DSCP 字段来提高丢弃的优先级，也可能会要求设备支持漏桶或令牌桶对特定数据包的发送速率进行限制。

由此可见，交换机需要支持的功能集合是随着空间和时间的变化而变化的，用模块化的思维构造交换流程能够快速定位需求差异并建立原型架构。

3.2　设计思想

RESSP 的设计思想是利用 FPGA 的可编程性将报文处理流水线的每个步骤都变得可重构，具体思路如下。

(1)RESSP 架构将报文的硬件转发流程划分为平台相关逻辑和平台无关逻辑。平台相关逻辑包括报文收发通道和异构平台配置通道，这与具体功能实现无关，对每个硬件而言是固定不变的。而平台无关逻辑即为可重构流水线。

(2)RESSP 将可重构流水线拆解成 4 个带有接口的独立功能模块，使用者可以根据实际使用需求自由选择流水线的每个模块，再对相应的文件进行编译，建立一个可重构的报文处理机制，以适应需求不一的网络应用场景，从而提升交换机的灵活性和可扩展性。

(3)RESSP 允许开发人员添加新模块，只需要使用标准定义的接口即可与其他模块连接交互，从而大大缩短了开发周期，适合于时间优先的原型设计方式，能够应对复杂变化的网络协议标准和定制的需求。

(4)RESSP 由于根据功能需求动态加载不同的处理模块，因此不会存在模块运行冗余或利用率低的问题，可以有效降低设备功耗，延长使用周期。

3.3　工作模型

RESSP 架构由可重构流水线库 Library 和报文缓存模块

Buffer 组成，如图 1 所示。

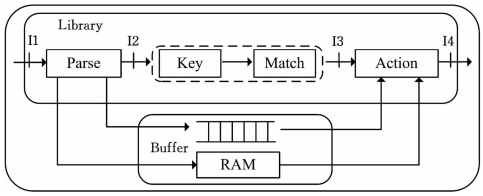


图 1 RESSP 架构

Fig. 1 The framework of RESSP

报文进入 RESSP 后，一边用 Buffer 存储报文(将报文存入 RAM 表，再将地址作为 Buffer_id 存入 FIFO)，一边提取报文的关键字，生成规则信息。规则信息在报文尾到达前生成完毕，动作模块继而将报文从 Buffer 中取出并根据规则信息对报文进行处理后再输出。

可重构流水线库是 RESSP 架构的核心组成部分，由以下 3 部分组成。

(1)4 类功能模块：Parse，Key，Match 和 Action。

Parse 代表报文解析模块，负责根据数据包的类型进行解析；Key 代表关键字提取模块，负责报文关键字的选择和组合；Match 代表匹配模块，用于匹配控制器下发的表项；Action 代表动作执行模块，指示对数据包的处理。4 类功能模块依次相连，代表一条基本的报文处理流水线。

(2)每类功能模块都包含有一个实现库。

Lib_Parse={Parse 1，Parse 2，Parse 3，…，Parse $N1$}；

Lib_Key={Key 1，Key 2，Key 3，…，Key $N2$}；

Lib_ Match={Match 1，Match 2，Match 3，…，Match $N3$}；

Lib_Action={Action 1，Action 2，Action 3，…，Action $N4$}。

Lib_Parse 代表 Parse 模块的实现库。据统计，对于一个数据包而言，携带 8 个或 8 个以上的头部信息很常见[13]，不同数据包之间的长度和格式差异很大，必须有针对性地进行识别并层层剥离头部以获得报文类型的信息。

Lib_Key 代表 Key 模块的实现库。根据不同类型的流的处理情况，提取需要用到的关键字组成匹配域。一条流水线可以选择多个 Key 模块。

Lib_Match 代表 Match 模块的实现库。尽管前一个 Key 模块决定了 Match 模块的选择，但相同字段的 Match 表可以采用不同的查表算法，以适应不同规模的网络应用场景。

Lib_Action 代表 Action 模块的实现库。一个 Action 代表的动作可以分成几个子模块完成。

实现库是相同功能模块的集合。每个模块在 verilog 代码中都拥有一个.v文件，使用者在选择某个模块时，就编译对应的.v文件。同时，允许向每个库中增加新模块或修改原有模块，以适应开发者的需求。

(3)4 组接口：I1，I2，I3 和 I4。

同一个实现库中的模块需要使用一组相同的接口，即与上(下)模块相连时所需要的传递的信号，新增加的模块通过标准接口与其他模块交互。

Key 模块和 Match 模块之间的信号交互由用户根据需要进行自定义，作为一个整体只要求满足 I2 和 I3 接口的规范即可。

图 2 给出了目前 RESSP 可重构流水线库的规划情况，从每个库中选择所需的功能模块进行编译，即可满足许多应用场景的转发需求。不同的用例可以挑选不同的模块组合成不同的处理流程。Use case 就是 RESSP 架构最基本的应用模式案例，代表层叠网中报文的处理流程。

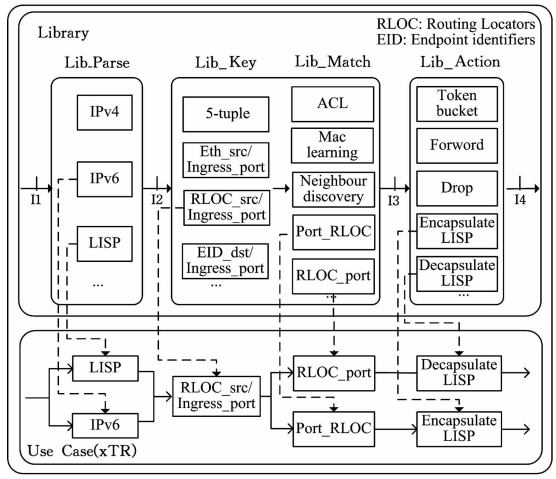


图 2 RESSP 架构的可重构流水线库

Fig. 2 Reconfigurable pipeline library of the framework of RESSP

4 RESSP 的实现与应用

本文基于实验平台 NetMagic08，在一个 SDN 交换机的原型 MiniSwitch 上对 RESSP 架构进行实现。本节首先对 NetMagic08 平台进行简要介绍，然后描述 MiniSwitch 系统的模块和功能实现，最后介绍基于 RESSP 的开源项目情况。

4.1 RESSP 实现平台

NetMagic08 是专门为网络创新研究定制的盒状的可重构开源实验平台，现在已被广泛用于网络研究和网络教学活动。NetMagic 采用转发逻辑与控制逻辑分离的设计思想，基于 FPGA 提供硬件的可编程性，确保了实验平台的可扩展性和可移植性[8]。

如图 3 所示，在控制器与交换机之间有一个交换机管理软件，它提供代理转换的功能，向上支持 OpenFlow 协议，解析上层控制器的配置管理命令，再通过读写虚拟地址空间的形式将其传递给硬件。考虑到 NetMagic 的兼容性，目前沿用 NMAC 控制协议[8]。因此，管理软、硬件交换机整体可以看作是 SDN 的数据平面，能够实现 SDN 环境的部署。

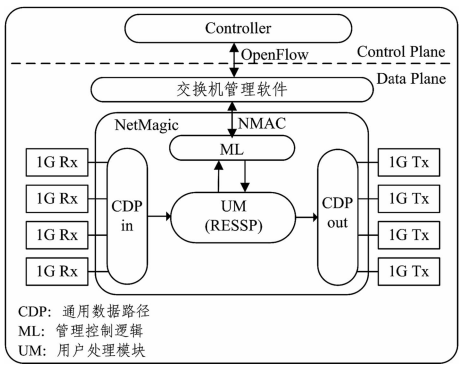


图 3 基于 NetMagic08 实现的 RESSP 系统模块图

Fig. 3 System module diagram of RESSP based on NetMagic08

RESSP 架构搭建在 NetMagic08 的 UM 模块中，负责报

文的基本处理流程。RESSP 每次只能基于一条选定的流水线运行，可根据应用环境的改变而重新选择或改良流水线模块的内容，从而编译出不同的案例，达到可编程网络的要求。

4.2　SDN 交换机原型 MiniSwitch

MiniSwitch 采用 RESSP 架构，是一个具有 SDN 基本功能的交换机原型，能够很好地支持模块化设计。

MiniSwitch 将寄存器、存储器和计数器等资源抽象出来，将类似于计算机内部的“虚拟地址空间”提供给管理软件；通过对这些虚拟地址空间的读、写等命令或函数，管理软件就可以访问到所有被管理的资源，并提取出映射内容，从而对 MiniSwitch 进行管理控制；MiniSwitch 组织这些虚拟地址空间形成转发策略。

在 RESSP 架构下，MiniSwitch 选用几种典型的数据包类型，如 IPv4 格式下的 ARP，TCP，UDP 和 ICMP 报文。Key 模块分别提取源 MAC 地址和 IP 五元组字段来匹配 Mac Learning 表和 ACL 表。Action 模块则提供令牌桶和链路仿真两种处理动作。其 RESSP 流水线模块的组合如图 4 所示。

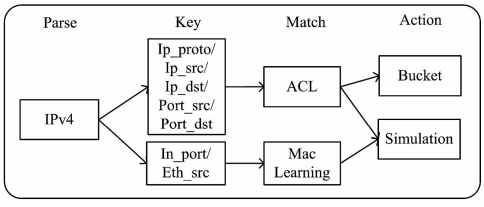


图 4　MiniSwitch 流水线

Fig. 4　Pipeline of MiniSwitch

表 1 列出了硬件代码在 QuartusII 中运行显示的资源使用情况，表 2 列出了为流水线模块的资源利用率列表。NetMagic08 FPGA 器件采用 ArriaIIGX EP2AGX45DF25C4，其含有 36100ALUTs 和 2939904Memory Bits。

表 1　MiniSwitch 资源使用情况

Table 1　The utilization of MiniSwitch resources

	占用资源	占用比例/%
Combinational ALUTs	16652	46.1
Dedicated logic registers	25342	70.2
Block Memory Bits	1607024	54.7

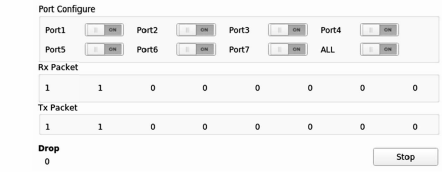
表 2　MiniSwitch 流水线资源使用情况

Table 2　The utilization of MiniSwitch pipeline resources

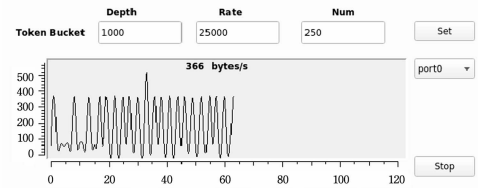
Model	Combinational ALUTs	Dedicated logic registers	Block memory bits
ArriaIIGX	36,100	36,100	2,939,904
Parse	307(0.85%)	433(1.2%)	35584(1.2%)
Key	438(1.2%)	511(1.4%)	464(0.02%)
Match	35(0.1%)	29(0.1%)	6720(0.2%)
Action	700(1.9%)	635(1.8%)	76800(2.6%)

从表中可以看出，流水线模块占用的 Netmagic08 总体逻辑资源低于 5%，故流水线各功能模块库有较大的扩充空间；RESSP 设计模型具有较低的硬件资源消耗量，适用于 SDN 数据平面的设计与扩展。而基于 NetMagic08 实现的 MiniSwitch原型可以满足一般的教学实验要求。下一步计划在多核同台和多核平台＋FPGA 上实现 RESSP 模型，以建立获取更复杂的网络功能。

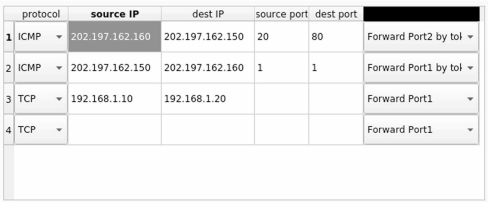
本文设计了一个交换机管理软件来验证 MiniSwitch 的正确性和有效性。图 5(a)所示的界面显示了对每个端口的开关控制和每个端口的流量统计情况，模拟了 SDN 控制器对物理网络的集中管理和流量控制。图 5(b)所示的界面通过设置令牌桶的桶深、令牌发送速率和一次发送的令牌个数来实现不同流的 QoS，通过后台线程对端口速率的统计，显示出通过令牌桶后流量变化的波形图，可验证 SDN 控制器基于流的 QoS 保障功能。图 5(c)给出了流表下发控制界面，通过用户的填写和选择来下发或删除 ACL 表项，从而实现对硬件 ACL 表的访问配置，体现控制器通过部署流表来指导数据平面按照相应规则转发的过程。



(a)端口流量统计



(b)设置令牌桶的效果



(c)流表下发控制

图 5　MiniSwitch 的图形化界面

Fig. 5　Graphical interface of MiniSwitch

MiniSwitch 为研究者观测网络内部的分组处理行为提供了一个良好的学习平台，已运用在国防科技大学、湖南大学等高校的网络实验课程中，加深了学生对网络交换、网络协议和网络管理的基本原理的认识，同时按照 RESSP 的规范进一步扩充了功能模块。

同时，高校研究人员也可以很方便地在 MiniSwitch 上进行功能扩充，搭建各种网络应用模型。东南大学的研究生利用该模型，通过对 Action 库的扩充，在 FPGA＋多核 CPU 的异构平台上加入精准发送模块，利用硬件的时间精确性，按照背靠背报文的间隔时间精准发送，以减小网络主动测量的误差。

5　进一步讨论

5.1　多级流表的支持

OpenFlow 在 1.1 版本后引入了多级流表(Multiple Flow Tables，MFT)[19]的概念，解决了单表尺寸臃肿的问题，这种改变直接增加了交换机的设计和实现难度，并降低了处理速度。虽然 MiniSwitch 仅反映了单表匹配的案例，但 RESSP 架构在支持 MFT 方面仍然有如下两种可行的实现方法。

(1)采用 flow cache[20]的设计思路，将命中率高的表项以单表形式放在 flow cache 中，以提高查表的速率。报文从硬

件端口进入，首先查找 flow cache，若未命中则送至交换机管理软件以进行 OpenFlow 模式下的多表匹配查询，若再不命中就发送至控制器以进行下一步处理。

(2)利用硬件实现 MFT，通过增加和自定义 Match 和 Match 模块的接口来实现数据分组在多张流表之间跳转，完成不同情况的匹配，如图 6 所示。

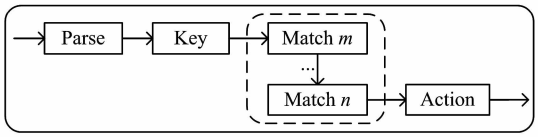


图 6 多表匹配的复杂组合

Fig. 6 Complex combination of multiple table matching

以上两种实现逻辑从软件和硬件两个角度完成了对 MFT 的支持，用户可以根据实际情况选择其中一种方式实现。

5.2 开发方法

对于需要采用 RESSP 架构来实现 SDN 交换机的研究人员来说，建议依次完成以下步骤后再进行代码的编写与调试。

(1)组合流水线。按照 RESSP 流水线的模式将转发策略细分成 4 个相应模块，可以在 FAST 开源社区中公开的可重构流水线库 lib/pipeline 中选择相似模块，也可自行开发新模块。

(2)定义模块间的接口。本实验使用的接口规范能够适应当前部分应用场景，是一种指导方法。可以沿用该接口规范完成部分应用场景的转发策略。但面对今后新协议或新应用的出现，可以充分利用 FPGA 的可编程性，参照原有设计思路，改良某些信号或位宽。

(3)合理地划分虚拟地址空间。交换机管理软件与硬件之间的交互，是利用中间控制协议对虚拟地址空间行操作的。因此，合理划分虚拟地址空间是设计中重要的一环。用户可以参考 MiniSwitch 的虚拟地址空间进行设计，首先划分 Match 表、Action、端口状态管理、计数器等基本功能的地址，再按照需求向下扩展。

结束语 针对现有交换机方案对 SDN 数据平面的教学科研支持不足的问题，本文提出了一种基于 FPGA 的可重构交换架构 RESSP。利用 RESSP 架构可以对报文转发流程进行调整和修改，以满足特定的网络需求，增强转发平面的灵活性和可扩展性。同时，实现了基于 RESSP 的 SDN 交换机原型 MiniSwitch。

通过开源所有设计文档和代码[9]，RESSP 架构将支持更多应用案例和创新研究，从而进一步推动 SDN 交换技术的发展，营造和提升中国高校的开源文化氛围。下一步的计划是构建一个软、硬件皆可编程的开放架构，在 FPGA＋多核 CPU 的异构平台上进行数据平面的设计。

致谢 感谢张洁和课题组成员毛健彪、徐东来给予的指导。

参 考 文 献

[1] NADEAU T D, GRAY K. SDN: Software DeRESSPd Networks [M]. O'Reilly Media, Inc, 2013.

[2] BOSSHART P, GIBBZ G, KIMY H S, et al. Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 99-110.

[3] Intel Ethernet Switch FM6000 Series[EB/OL]. http://www.intel.com/content/www/us/en/ethernet-products/switch-silicon/ethernet-switch-fm5000-fm6000-series.html.

[4] 电子发烧友[OL]. http://www.elecfans.com/emb/fpga/20150907382835.html.

[5] NAOUS J, ERICKSON D, COVINGTON G A, et al. Implementing an OpenFlow switch on the NetFPGA platform[C]// Proceeding of the 4th ACM/IEEE Symposium on Architectures for Networking and Communication System. 2008: 1-9.

[6] HU C C, YANG J, GONG Z M, et al. DesktopDC: Setting All Programmable Data Center Networking Testbed on Desk [J]. ACM SIGCOMM Computer Communication Review, 2014, 44(4): 593-594.

[7] JIA C B, HUANG J F, SU Q, et al. OpenFlow Implementation on NetMagic Platform [J]. Applied Mechanics & Materials, 2012, 198-199: 516-522.

[8] CAO C Z, MAO J B, SUN Z G, et al. Method of NetMagic hardware development[J]. Computer Engineering & Science, 2014, 36(9): 1678-1683. (in Chinese)
曹成周，毛健彪，孙志刚，等. NetMagic 平台硬件开发方法[J]. 计算机工程与科学，2014，36(9)：1678-1683.

[9] FAST[OL]. http://fast-switch.github.io.

[10] ONF. OpenFlow Table Type Pattrns Version 1.0[S]. 2014.

[11] JAIN S, KUMAR A, MANDAL S, et al. B4: experience with a globally-deployed software defined wan[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 3-14.

[12] BOSSHART P, DALY D, GIBBY G, et al. P4: Programming Protocol-Independent Packet Processors[J]. Computer Communication Review: A Quarterly Publication of the Special Interest Group on Data Communication. 2014, 44(3): 88-95.

[13] SONG H Y. Unaware Routing Protocol within OpenFlow2.0 [J]. Communications of the CCF, 2015, 1(11): 35-41.

[14] The world's fastest and most programmable networks [EB/OL]. https://barefootnetworks.com/media/white_papers/Barefoot-Worlds-Fastest-Most-Programmable-Networks.pdf.

[15] GIBB G, VARGHESE G, HOROWITZ M, et al. Design Principles for Packet Parsers[C]// 2013 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS). 2013: 13-24.

[16] DNLAB[OL]. http://www.sdnlab.com/1710.html.

[17] TECHCON A. Xilinx Introduces SDNet & 'Softly' DeRESSPd Networks [OL]. http://www.xilinx.com/products/design-tools/software-zore/sdnet.html.

[18] Xilinx SDNet: A New Way to Specify Network Hardware[EB/OL]. http://www.xilinx.com/publications/prod_mktg/linley-group-sdnet-wp.pdf.

[19] Specification, OpenFlow Switch. Version 1.1.0 Implemented (Wire Protocol 0x02)[S]. 2011.

[20] SHELLY N, JACKSON E J, KOPONEN T, et al. Flow Caching for High Entropy Packet Fields[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(4): 151-156.